

## SRM v2.2 WLCG usage agreement

SRM v2.2 WLCG usage agreement.....	1
1 Purpose of this document.....	3
2 WLCG usage agreement SRM v2.2 specification .....	3
2.1 Storage classes .....	3
2.2 Removal policies.....	3
2.3 Protocol negotiation .....	4
2.4 Information discovery.....	4
2.5 srmReserveSpace .....	4
2.5.1 parameters .....	4
2.5.2 WLCG interpretation .....	5
2.5.3 Usage restrictions.....	5
2.6 srmLs.....	5
2.6.1 parameters .....	5
2.6.2 WLCG usage.....	6
2.6.3 Usage restrictions.....	6
2.6.4 Caveat .....	6
2.7 srmPrepareToGet .....	6
2.7.1 parameters .....	6
2.7.2 WLCG interpretation .....	7
2.7.3 Usage restrictions.....	7
2.8 srmPrepareToPut.....	7
2.8.1 parameters .....	7
2.8.2 WLCG interpretation .....	8
2.8.3 Usage restrictions.....	8
2.9 srmCopy .....	9
2.9.1 parameters .....	9
2.9.2 WLCG interpretation .....	9
2.9.3 Usage restrictions.....	9
2.10 srmChangeSpaceForFiles .....	10
2.10.1 parameters .....	10
2.10.2 WLCG interpretation .....	10
2.10.3 Usage restrictions.....	10
2.11 srmRm.....	10
2.11.1 parameters .....	10
2.11.2 WLCG interpretation .....	11
2.11.3 Usage restrictions.....	11
2.12 srmPurgeFromSpace .....	11
2.12.1 parameters .....	11
2.12.2 WLCG interpretation .....	11
2.12.3 Usage restrictions.....	11
2.13 srmBringOnline.....	11
2.13.1 parameters .....	11
2.13.2 WLCG interpretation .....	12
2.13.3 Usage restrictions.....	12



# 1 Purpose of this document

This note summarizes the agreed client usage and server behaviour for the SRM v2.2 implementations used by WLCG applications. The agreement encompass the clients

- FTS
- GFAL
- LCG-util

and storage providers:

- dCache
- DPM
- CASTOR

The agreement shall be focused on meeting the LHC experiments' requirements for grid storage interfaces. It is understood that significant changes to the SRM v2 specifications are not feasible at the timescale for LHC production. The note therefore specifies how the various existing methods and data objects, as they are specified in the agreed v2.2 specification, should be used in order meet the LHC requirements. The outlined WLCG interpretation of the interfaces may be suggest more restrictive use of the specification than other implementations of SRM client and servers although care is taken to preserve interoperability with the latter.

## 2 WLCG usage agreement SRM v2.2 specification

### 2.1 Storage classes

Wherever usage of storage class argument is agreed to be required by WLCG, the agreed terminology is:

- TAccessLatency {ONLINE, NEARLINE}
- TRetentionPolicy {REPLICA, CUSTODIAL}

The 'OUTPUT' value of the TRetentionPolicy as well as the 'OFFLINE' value of the TAccessLatency shall *not* be used by any WLCG application. The mapping to labels 'TapeXDiskY' is given by:

- Tape1Disk0: NEARLINE + CUSTODIAL
- Tape1Disk1: ONLINE + CUSTODIAL
- Tape0Disk1: ONLINE + REPLICA

WLCG clients shall only use those three combinations of TAccessLatency and TRetentionPolicy.

In all subsequent methods listed in the following, the allowed/disallowed usage of storage classes attributes must be specified.

### 2.2 Removal policies

All WLCG files are 'permanent' meaning that any implicit automatic removal of the file from the storage system is disallowed. The file lifecycle within an SRM is always terminated by the user calling the srmRm method.

## 2.3 Protocol negotiation

The protocols are specified in preferred order on all srm transfer functions (get, put, bringOnline). This is already implicit (or explicit?) in the srm specification.

## 2.4 Information discovery

Process to achieve agreement:

- srmGetStorageInfo shall be removed from the specification. There is no requirement for an SRM information interface for LCG production
- Client implementations provide list of required information
- GLUE defines the schema accordingly
- In the longer term implementations may provide the interface to query for the information required for the GLUE schema

The vehicle for achieving the agreement is the weekly wlcg-dm phone conference.

## 2.5 srmReserveSpace

### 2.5.1 parameters

In:	TUserID String TRetentionPolicyInfo TTransferParameters TSizeInBytes TSizeInBytes TLifeTimeInSeconds Int [] TStorageSystemInfo	authorizationID, userSpaceTokenDescription, preferredRetentionPolicyInfo, transferParameters, sizeOfTotalSpaceDesired, <u>sizeOfGuaranteedSpaceDesired</u> , lifetimeOfSpaceToReserve, expectedFileSize, storageSystemInfo
Out:	TRequestToken TLifeTimeInSeconds TRetentionPolicyInfo TSizeInBytes TSizeInBytes TLifeTimeInSeconds TSpaceToken, TReturnStatus	requestToken estimatedProcessingTime, retentionPolicyInfo, sizeOfTotalReservedSpace, // best effort sizeOfGuaranteedReservedSpace, lifetimeOfReservedSpace, spaceToken, <u>returnStatus</u>
typedef	struct {TAccessPattern TConnectionType string[] string[] } TTransferParameters	accessPattern, connectionType, arrayOfTransferProtocols, arrayOfClientNetworks

### **2.5.2 WLCG interpretation**

For practical use srmReserveSpace only deals with space reservations on disk cache while tape space is always considered infinite. SRM implementations may allow dynamic space reservations for any user while other implementation may only allow static reservations restricted for use by certain VO roles. For static reservations details requirements for the reservation, like network connectivity, may be negotiated directly between storage provider and client.

### **2.5.3 Usage restrictions**

- Clients shall only specify sizeOfGuaranteeSpaceDesired while sizeOfTotalSpaceDesired shall be ignored by all servers.
- Server implementations are allowed to restrict permission to reserve space to certain VO roles. The client shall handle the corresponding error:  
SRM\_AUTHORIZATION\_FAILURE
- The expectedFileSize shall not be used.
- The preferredRetentionPolicyInfo should be renamed to retentionPolicyInfo. The corresponding output argument shall be removed. If the requested retentionPolicyInfo cannot be satisfied SRM\_INVALID\_REQUEST error code shall be returned.
- Clients shall not specify storageSystemInfo and servers shall ignore any value provided in this parameter.
- SRM implementation may return infinite lifetime no matter what was specified by the user. It is then the client responsibility to remove the space.
- The TTransferProtocol data structure shall be renamed to TTransferParameters in the SRM specification
- A transferParameter input parameter shall be added to the srmReserveSpace method in the SRM specification
- WLCG clients shall use srmGetSpaceTokens method with the user space description to get to the space token
- WLCG clients shall not use dynamic space reservations initially
- The transferParameter is *optional*

## **2.6 srmLs**

### **2.6.1 parameters**

In:	TUserID	authorizationID,
	TStorageSystemInfo	storageSystemInfo,
	TSURL []	<u>surls</u> ,
	TFileStorageType	fileStorageType,
	boolean	fullDetailedList,
	boolean	allLevelRecursive,
	int	numOfLevels,
	int	offset,
	int	count

Out: `TMetaDataPathDetail[]` `details,`  
`TReturnStatus` `returnStatus`

## 2.6.2 WLCG usage

Client use srmLs for getting metadata attributes of individual files in the same way as getFileMetaData was used in srm v1.1. Support for directory listings can be motivated for cross-checking file catalogues.

### 2.6.3 Usage restrictions

- TFileLocality {ONLINE, NEARLINE, ONLINE\_AND\_NEARLINE, LOST, NONE, UNAVAILABLE} attribute of the srmLs output shall be supported for individual file queries but not for directory listings even if detailed directory listing has been requested.
  - spaceToken attribute of the srmLs output shall be supported for individual file queries but not for directory listings even if detailed directory listing has been requested.
  - The LOST value of TFileLocality shall be used if the file is lost because of a permanent hardware failure. The NONE value shall be used if the file is empty (zero size). The UNAVAILABLE value shall be used if the file is unavailable due to temporary hardware failure.
  - For directory listings srmLs output may be constrained to an upper limit of the number of entries returned.
  - srmLs for a non-existing file or directory shall result in an SRM\_INVALID\_PATH error
  - If numberOfLevels is specified to zero the returned information shall be for the directory itself and not the files in the directory.

#### **2.6.4 Caveat**

Support for offset and count in the srmLs function at the timescale for LHC production will require more discussion. Currently no WLCG implementation supports it.

## 2.7 *srmPrepareToGet*

## 2.7.1 parameters

In:	TUserID TGetFileRequest[] string TStorageSystemInfo TLifeTimeInSeconds TFileStorageType TLifeTimeInSeconds TSpaceToken TRetentionPolicyInfo TTransferProtocol	authorizationID, <u>arrayOfFileRequest</u> , userRequestDescription, storageSystemInfo, totalRetryTime preferredFileStorageType desiredLifetime, targetSpaceToken targetFileRetentionPolicyInfo transferProtocolList
-----	--	---

Out:	TRequestToken	requestToken,
	TReturnStatus	<u>returnStatus</u> ,
	TGetRequestFileStatus[]	arrayOfFileStatus

## 2.7.2 WLCG interpretation

srmPrepareToGet is used for preparing a file for transfer or access. WLCG usage exclude usage srmPrepareToGet for changing space or retention attributes of the file.

## 2.7.3 Usage restrictions

- WLCG tools/clients shall not use space tokens, or retention policy/access latency on a srmPrepareToGet
- Transfer protocol is mandatory (underlined above but not yet in the specification)
- srmPrepareToGet is asynchronous (specifications should change)
- It is up to the storage system to decide the TConnectionType if not provided.
- WLCG clients shall not have expectations of any value returned for the estimated\*Time.
- fileStorageType and spaceToken shall be removed from TGetFileRequestStatus in the SRM specification.
- If any of the requested files is temporarily unavailable (e.g. due to hardware failure) an error SRM\_FILE\_UNAVAILABLE is returned
- If any of the request files is permanently lost (e.g. due to hardware failure) an error SRM\_FILE\_LOST is returned
- The SRM implementation should fail with an error SRM\_FILE\_BUSY a srmPrepareToGet request for files for which there is an active srmPrepareToPut (no srmPutDone not yet called).
- The totalRetryTime shall be renamed totalRequestTime in the SRM specification. If the request does not complete within the totalRequestTime, the request will fail with SRM\_TIMEDOUT. The totalRequestTime is negotiated with the SRM implementation. The totalRequestTime attribute should be added to the output parameters of the srmStatusOfGetRequest method. This functionality is desired for the longer term but *not* required for the start of LCG production
- The WLCG SRM implementations shall allow srmPrepareToGet request to continue provided that at least one file is successful or in progress.

## 2.8 srmPrepareToPut

### 2.8.1 parameters

In:	TUserID	authorizationID,
	TPutFileRequest[]	<u>arrayOfFileRequest</u> ,
	string	userRequestDescription,
	TOverwriteMode	overwriteOption,
	TStorageSystemInfo	storageSystemInfo,
	TLifeTimeInSeconds	totalRetryTime
	TLifeTimeInSeconds	desiredPinLifetime, // on TURL
	TLifeTimeInSeconds	desiredFileLifetime, // on SURL

TFileStorageType	preferredFileStorageType,
TSpaceToken	targetSpaceToken
TRetentionPolicyInfo	targetFileRetentionPolicyInfo
TTransferProtocol	<u>transferProtocolList</u>
Out:	
TRequestToken	requestToken,
TReturnStatus	<u>returnStatus</u> ,
TPutRequestFileStatus[]	arrayOfFileStatus

## 2.8.2 WLCG interpretation

...

## 2.8.3 Usage restrictions

- Transfer protocol is mandatory (underlined above but not yet in the specification)
- srmPrepareToPut is asynchronous (specifications should change)
- It is up to the storage system to decide the TConnectionType if not provided.
- WLCG clients shall not have expectations of any value returned for the estimated\*Time
- fileStorageType and spaceToken shall be removed from TPutFileRequestStatus in the SRM specification.
- transferProtocolInfo shall be changed to be an array of key-value pairs. The parameter will initially not be supported for WLCG production
- The WLCG client shall provide SURLs and hence SRM implementations are not required to generate names
- The totalRetryTime shall be renamed totalRequestTime in the SRM specification. If the request does not complete within the totalRequestTime, the request will fail with SRM\_TIMEDOUT. The totalRequestTime is negotiated with the SRM implementation. The totalRequestTime attribute should be added to the output parameters of the srmStatusOfPutRequest method. This functionality is desired for the longer term but *not* required for the start of LCG production
- The WLCG SRM implementations shall allow srmPrepareToPut request to continue provided that at least one file is successful or in progress
- If both targetSpaceToken and targetFileRetentionPolicyInfo are provided the retention policy attributes must match that of the specified space. Otherwise the request fail with SRM\_INVALID\_REQUEST
- WLCG clients shall use the targetSpaceToken and *not* the targetFileRetentionPolicyInfo. If the targetSpaceToken is not provided a default space will be chosen by the SRM.
- When a SURL lifetime is added to the SRM specification (for non-WLCG implementations) it shall not be used by WLCG clients and WLCG SRM implementations are not required to support it.
- The TURL lifetime is the allowed time during which the file may be written; there is no guarantee as to the availability of the TURL for writing after this time.
- A TURL returned by prepareToPut may not be used for read access with any protocol. An explicit srmPrepareToGet/srmBringOnline call is required.

- The SRM implementation should fail with an error SRM\_FILE\_BUSY a srmPrepareToPutrequest for files for which there is an active srmPrepareToPut (no srmPutDone not yet called).

## 2.9 *srmCopy*

### 2.9.1 parameters

In:	TUserID TCopyFileRequest[] string TOverwriteMode Boolean TLifeTimeInSeconds TLifeTimeInSeconds TFileStorageType TOverwriteMode TSpaceToken TRetentionPolicyInfo	authorizationID, <u>arrayOfFileRequest</u> , userRequestDescription, overwriteOption, removeSourceFiles (default = false), totalRetryTime lifetime, // on target SURLs targetFileStorageType, overwriteMode, targetSpaceToken, targetFileRetentionPolicyInfo
Out:	TRequestToken TReturnStatus TCopyRequestFileStatus[]	requestToken, <u>returnStatus</u> , arrayOfFileStatus

### 2.9.2 WLCG interpretation

...

### 2.9.3 Usage restrictions

- The dirOption (in TCopyFileRequest) will not be supported initially
- fileStorageType and spaceToken shall be removed from TCopyFileRequestStatus in the SRM specification
- The SURL lifetime shall not be used by WLCG clients and WLCG SRM implementations are not required to support it.
- Files are immutable. WLCG clients shall not specify overwriteOption. If SRM returns failure, the state of the system shall be as if the transfer did not take place.
- The removeSourceFiles flag shall be removed from the specification.
- The totalRetryTime shall be renamed totalRequestTime in the SRM specification. If the request does not complete within the totalRequestTime, the request will fail with SRM\_TIMEDOUT. The totalRequestTime is negotiated with the SRM implementation. The totalRequestTime attribute should be added to the output parameters of the srmStatusOfCopyRequest method. This functionality is desired for the longer term but *not* required for the start of LCG production.
- WLCG clients shall use the targetSpaceToken and *not* the targetFileRetentionPolicyInfo. If the targetSpaceToken is not provided a default space will be chosen by the SRM.

- The SURLs passed in a call to the srmAbortFiles method following a srmCopy are the target SURLs (which are bound to be unique).
- The SRM implementation should fail with an error SRM\_FILE\_BUSY a srmPrepareToCopyRequest for files (both source and target) for which there is an active srmPrepareToPut (no srmPutDone not yet called).

## **2.10 srmChangeSpaceForFiles**

### **2.10.1 parameters**

In:	TUserID TSURLInfo[] TSpaceToken	authorizationID <u>arrayOfSURLs</u> <u>targetSpaceToken</u>
Out:	TRequestToken TLifeTimeInSeconds TReturnStatus	requestToken estimatedProcessingTime <u>returnStatus</u>

### **2.10.2 WLCG interpretation**

The common use case is the transition Tape1Disk1 → Tape1Disk0 or vice versa. Change of retention policy Tape0Disk1 → Tape1Disk[0-1]???

### **2.10.3 Usage restrictions**

- srmChangeSpaceForFiles shall be added to the SRM specification
- srmChangeSpaceForFiles is asynchronous
- For any transition forbidden by the SRM implementation, an error SRM\_INVALID\_SPCTRANSITION is returned
- srmChangeSpaceForFiles is not required for the start LCG production but short after
- WLCG SRM implementations shall not change the SURL when a file is moved to a new space
- A srmStatusOfChangeSpaceForFiles method shall be added to the SRM specification. The signature of this functions remains to be discussed between the SRM implementors
- **WLCG SRM implementors and clients shall decide which transitions should be supported**

## **2.11 srmRm**

### **2.11.1 parameters**

In:	TUserID TSURLInfo[]	authorizationID, <u>arrayOfSURLs</u>
Out:	TReturnStatus TSURLReturnStatus[]	<u>returnStatus</u> , <u>arrayOfFileStatus</u>

## **2.11.2 WLCG interpretation**

Removes name space entry and all associated copy of the file.

## **2.11.3 Usage restrictions**

Clients shall or shall not....

## **2.12 srmPurgeFromSpace**

### **2.12.1 parameters**

In:	TUserID TSURLInfo[] TSpaceToken	authorizationID <u>arrayOfSURLs</u> <u>spaceToken</u>
Out:	TReturnStatus TSURLReturnStatus[]	<u>returnStatus,</u> arrayOfFileStatus

## **2.12.2 WLCG interpretation**

Purges files from the given space. Difference to srmReleaseFiles and srmAbortFiles is that srmPurgeFromSpace is not associated with a request. srmReleaseFiles is enhanced with a flag ‘remove’ giving a hint for the immediate garbage collection of disk copies.

## **2.12.3 Usage restrictions**

- If the specified space contains the only remaining copy of the file in the storage system, an error SRM\_LAST\_COPY is returned.
- The method shall only succeed if there are no outstanding pins or requests for the specified files. Otherwise an error SRM\_FILE\_BUSY is returned.

## **2.13 srmBringOnline**

### **2.13.1 parameters**

In:	TUserID TGetFileRequest[] string TStorageSystemInfo TLifeTimeInSeconds TFileStorageType TLifeTimeInSeconds TSpaceToken TRetentionPolicyInfo TTransferProtocol	authorizationID, <u>arrayOfFileRequest,</u> <u>userRequestDescription,</u> storageSystemInfo, totalRetryTime preferredFileStorageType desiredLifetime, // pin time targetSpaceToken, targetFileRetentionPolicyInfo <u>transferProtocolList</u>
Out:	TRequestToken TReturnStatus	<u>requestToken,</u> <u>returnStatus,</u>

```
string[] supportedTransferProtocols,  
TBringOnlineRequestFileStatus[] arrayOfFileStatus
```

## 2.13.2 WLCG interpretation

### 2.13.3 Usage restrictions

- srmBringOnline is asynchronous (specifications should change)
- Protocol matching shall work like for srmPrepareToGet
- A deferredStartTime hint parameter shall be added to the specification for future use. The parameter means that the client does not intend to use the files before that time. The purpose is to be able to support CE – SE resource co-allocations and tape mounting efficiency.
- WLCG tools/clients shall not use space tokens, or retention policy/access latency on a srmBringOnline
- If Transfer protocol list is not specified the default is assumed to be the {processing mode, LAN access} for the site. WLCG clients are strongly (please...) recommended to specify the transfer protocols.
- It is up to the storage system to decide the TConnectionType if not provided
- WLCG clients shall not have expectations of any value returned for the estimated\*Time.
- fileStorageType and spaceToken shall be removed from TBringOnlineFileRequestStatus in the SRM specification.
- If any of the requested files is temporarily unavailable (e.g. due to hardware failure) an error SRM\_FILE\_UNAVAILABLE is returned
- If any of the request files is permanently lost (e.g. due to hardware failure) an error SRM\_FILE\_LOST is returned
- The SRM implementation should fail with an error SRM\_FILE\_BUSY a srmBringOnline request for files for which there is an active srmPrepareToPut (no srmPutDone not yet called).
- The totalRetryTime shall be renamed totalRequestTime in the SRM specification. If the request does not complete within the totalRequestTime, the request will fail with SRM\_TIMEDOUT. The totalRequestTime is negotiated with the SRM implementation. The totalRequestTime attribute should be added to the output parameters of the srmStatusOfBringOnline method. This functionality is desired for the longer term but *not* required for the start of LCG production
- The WLCG SRM implementations shall allow srmBringOnline request to continue provided that at least one file is successful or in progress